



PAPER

Compiling quantum algorithms for architectures with multi-qubit gates

OPEN ACCESS

RECEIVED
9 February 2016REVISED
2 May 2016ACCEPTED FOR PUBLICATION
2 June 2016PUBLISHED
24 June 2016

Original content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Esteban A Martinez^{1,3}, Thomas Monz¹, Daniel Nigg¹, Philipp Schindler¹ and Rainer Blatt^{1,2}¹ Institut für Experimentalphysik, Universität Innsbruck, Technikerstraße 25/4, 6020 Innsbruck, Austria² Institut für Quantenoptik und Quanteninformation, Österreichische Akademie der Wissenschaften, Technikerstraße 21a, 6020 Innsbruck, Austria³ Author to whom any correspondence should be addressed.E-mail: esteban.martinez@uibk.ac.at**Keywords:** quantum algorithms, quantum compiler, many-qubit entangling gates, Mølmer-Sørensen gates, trapped-ion quantum computing

Abstract

In recent years, small-scale quantum information processors have been realized in multiple physical architectures. These systems provide a universal set of gates that allow one to implement any given unitary operation. The decomposition of a particular algorithm into a sequence of these available gates is not unique. Thus, the fidelity of the implementation of an algorithm can be increased by choosing an optimized decomposition into available gates. Here, we present a method to find such a decomposition, where a small-scale ion trap quantum information processor is used as an example. We demonstrate a numerical optimization protocol that minimizes the number of required multi-qubit entangling gates by design. Furthermore, we adapt the method for state preparation, and quantum algorithms including in-sequence measurements.

1. Introduction

Quantum technologies open new possibilities that are inaccessible with current classical devices, ranging from cryptography [1, 2] to efficient simulation of physical systems [3–5]. To utilize the full computational power of quantum systems, one needs a *universal quantum computer*: a device able to implement arbitrary unitary operations, or at least to approximate them to arbitrary accuracy. However, in any specific physical system, only a certain set of operations is readily available. Therefore, it is necessary to decompose the desired unitary operation as a sequence of these experimentally available gates. An available set of gates is known as *universal* if it is possible to find such a decomposition for an arbitrary unitary quantum operation acting on the qubit register.

A canonical universal set of gates consists of two-qubit CNOT gates and arbitrary single-qubit rotations. There exist deterministic algorithms that provide near-optimal decompositions of unitaries in terms of these gates [6]. However, the set of gates that yields the highest fidelities depends on the particular experimental implementation. In particular, two-qubit CNOT gates may not be the most efficient to implement. Architectures like trapped ions [7, 8] or atom lattices [9] include in their toolboxes high-fidelity multi-qubit gates that act on the entire qubit register (see section 1.1). Implementing two-qubit gates in terms of these requires refocusing [10] or decoupling [7] techniques, and thus increases the overhead. Therefore it is desirable to find a direct decomposition of the target unitary into the available operations. In general, the number of multi-qubit gates needs to be minimized, since these are more prone to errors than local gates.

Compiling unitaries using multi-qubit gates that act on the whole qubit register is more challenging than using two-qubit gates. Even if a sequence implements correctly a unitary for N qubits, it might not work for $N + 1$ qubits, since additional ‘spectator’ qubits will also be affected by the sequence instead of being left unchanged [11]. Therefore, one has to define a qubit register of interest where the unitary will be compiled, and the experimental implementation of the resulting sequence has to be limited to this subregister, as explained in section 1.1. Moreover, the existing analytical methods for finding decompositions of unitaries in terms of two-

qubit gates (see for instance [12]) do not seem to apply to multi-qubit gates. Therefore, in this work we employ an approach based on numerical optimization.

A similar algorithm for finding multi-qubit gate decompositions has been studied in [11], where optimal control techniques are used to find a pulse sequence for a given target unitary operation. The procedure described there starts with long sequences and then removes pulses, if possible. This often results in sequences with more entangling operations than actually required. In this work we present an algorithm designed to produce decompositions with a minimal number of entangling gates. In addition, we introduce a deterministic algorithm for finding decompositions of local unitaries. We also extend the algorithm to operations required for state preparation or measurement, which are particular cases of more general operations known as *isometries* [13, 14].

The paper is organized as follows: in section 1.1 we describe precisely which gates we will consider as part of our experimentally available toolbox, and review some architectures for quantum information processing to which the methods described in this work can be applied. In section 2 we show an analytic algorithm to compile local unitaries, which can be used to find efficient implementations of state and process tomographies. Finally, in section 3 we describe and analyze an algorithm to compile fully general unitaries which relies on numerical optimization.

1.1. Experimental toolbox

Several quantum information processing experiments based on atomic and molecular systems have similar toolsets of quantum operations at their disposal. Often, it is convenient to apply collective rotations on an entire qubit register. These collective (yet local) gates, combined with addressed operations (typically rotations around the Z axis) allow one to implement arbitrary local unitaries, as we show in section 2. Together with suitable multi-qubit operations, arbitrary quantum unitaries can be implemented. In this work we consider the following set of gates:

- Collective rotations of the whole qubit register about any axis on the equator of the Bloch sphere $C(\theta, \phi)$. Here θ is the rotation angle and ϕ is the phase, so that:

$$C(\theta, \phi) = e^{-i\theta(S_x \cos \phi + S_y \sin \phi)/2}, \quad (1)$$

where $S_{x,y} = \sigma_1^{x,y} + \dots + \sigma_N^{x,y}$ are the total spin projections on the x or y axes, and $\sigma_j^{x,y,z}$ are the respective Pauli operators corresponding to qubit j . For the sake of brevity we also define rotations around the X and Y axes as:

$$X(\theta) = C(\theta, 0), \quad (2)$$

$$Y(\theta) = C(\theta, \pi/2). \quad (3)$$

- Single qubit rotations around the Z axis $Z_n(\theta)$, where θ is the rotation angle, and n is the qubit index:

$$Z_n(\theta) = e^{-i\theta\sigma_n^z/2}, \quad (4)$$

with σ_n^z being the Pauli Z operator applied to the n -th ion.

- Entangling Mølmer-Sørensen (MS) gates [15], with arbitrary rotation angle and phase $MS_\phi(\theta)$. Here θ is the rotation angle and ϕ is the phase of the gate, resulting in:

$$MS_\phi(\theta) = e^{-i\theta(S_x \cos \phi + S_y \sin \phi)^2/4}, \quad (5)$$

where $S_{x,y} = \sigma_1^{x,y} + \dots + \sigma_N^{x,y}$ are the total spin projections on the x or y axes, as before. For $\phi = 0$ or $\phi = \pi/2$ we obtain gates that act around the X or Y axes, which we will denote:

$$MS_{x,y}(\theta) = e^{-i\theta S_{x,y}^2/4}. \quad (6)$$

As mentioned before, it is desirable to be able to restrict the action of the MS gate to a particular qubit subset. This can be done experimentally by spectroscopically decoupling the rest of the qubits from the computation [7], or by addressing the MS gate only on the relevant subset of the qubits [16].

This set of gates, or equivalent ones, are available in several trapped-ion experiments [7, 17, 18]. A similar toolbox of operations is available for architectures based on trapped-ion hyperfine qubits. For example, [8] describes high-fidelity microwave gates applied to a single hyperfine $^{43}\text{Ca}^+$ qubit. In a multi-qubit system, these gates would drive collective rotations like the ones described above. In addition, [19] describes a Raman-driven $\sigma^z \otimes \sigma^z$ phase gate on two qubits, which applied to a many-qubit register would act analogously to the MS gate already described.

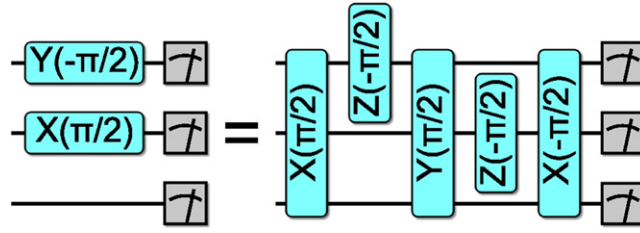


Figure 1. Pulse sequence to perform a projective measurement on the $\{X, Y, Z\}$ bases for qubits $\{1, 2, 3\}$, respectively.

Recently, an implementation of high-fidelity gates in a 2D array of neutral atom qubits was reported [9]. The toolbox described there consists of global microwave-driven gates and single-site Stark shifts on the atoms, which are completely equivalent to the local operations described before for the trapped-ion architecture. A multi-qubit CNOT gate, equivalent to the MS gate, could also be implemented by means of long-range Rydberg blockade interactions [20].

2. Compilation of local unitaries

Local unitaries can be written as a product of single-qubit unitaries. In this section we show a fully deterministic algorithm that produces decompositions of any local unitary as a sequence of collective equatorial rotations and addressed Z rotations, as described in section 1.1. The decompositions presented here are optimal in the number of pulses. These techniques are particularly useful for the implementation of state and process tomographies, as exemplified in figure 1, since both require only local operations at the beginning and end of the algorithm.

Let us consider a register of N qubits, and a local unitary $U = U_1 \otimes U_2 \otimes \dots \otimes U_N$ to be applied to them, where U_i is the action of the unitary on the i -th qubit. If the same operation has to be applied to more than one qubit ($U_i = U_j$), we can replace both with a single instance of the operation, and then apply the same addressed rotations on all qubits subject to the same operation U_i . Therefore, we only have to consider the case where every U_i is unique.

In order to apply a general local unitary to each qubit we need to have at least three degrees of freedom per qubit [6], so the decomposition must have at least $3N$ free parameters. During the sequence at least $N - 1$ of the qubits must eventually be addressed, since a different unitary has to be applied to each qubit. Therefore, a sequence of addressed operations of the form $Z_1(\theta_1), Z_2(\theta_2), \dots, Z_{N-1}(\theta_{N-1})$ must be included in the decomposition. These provide $N - 1$ parameters, so $2N + 1$ more degrees of freedom are required. The most economic way to provide these is by means of collective gates $C(\theta_i, \phi_i)$, which have two degrees of freedom each, so the shortest sequence possible must include at least N global operations, for a total of $3N - 1$ free parameters. One additional degree of freedom remains, so we must add a last gate. This can be either an addressed operation on qubit N or a collective gate. If we add an addressed gate Z_N , we obtain a sequence of the form:

$$U = Z_N C_N Z_{N-1} C_{N-1} \dots Z_2 C_2 Z_1 C_1, \quad (7)$$

where $C_i = C(\theta_i, \phi_i)$ and $Z_i = Z_i(\theta_i)$ are collective and single-qubit rotations respectively, as explained in section 1.1. Such a sequence is useful for compiling local unitaries up to arbitrary phases, as explained in appendices A.3 and A.4. The second alternative is to add a collective rotation C'_N :

$$U = C'_N C_N Z_{N-1} C_{N-1} \dots Z_2 C_2 Z_1 C_1, \quad (8)$$

which is the type of sequence we consider in this section.

For particular unitaries, some of the C_i and Z_i in equation (8) may actually be the identity, in which case the sequence is simpler. Since the decomposition depends on the ordering of the qubits, by reordering them a simpler sequence might be obtained. For small numbers of qubits, one can compile the unitary for every possible permutation, although this becomes inefficient for large numbers of qubits. However, let us remember that, for the purposes of the compilation, the qubits are grouped together according to which of them experience the same single-qubit unitary U_i . For an application such as state tomography, there are only three possible unitaries to be applied to each qubit in the register (shown in figure 1), since one only wants to perform a measurement in one of three different bases. Therefore, effectively we only need to consider three qubits, in which case trying out all the permutations is perfectly feasible.

We will describe now how to compile a generic local unitary $U = U_1 \otimes U_2 \otimes \dots \otimes U_N$ exactly, using a decomposition of the form (8). Let us first note that the unitaries in equation (8) act on the N -qubit-Hilbert

space, which is the tensor product of the single-qubit–Hilbert spaces. For the sake of simplifying the notation, we will now refer to these unitaries as \tilde{C}_i , \tilde{Z}_i , and will reuse the notations C_i and Z_i for their action on the single qubits, so that:

$$\begin{aligned}\tilde{C}_i &= C_i \otimes C_i \otimes \cdots \otimes C_i, \\ \tilde{Z}_i &= 1 \otimes 1 \otimes \cdots \otimes Z_i \otimes \cdots \otimes 1,\end{aligned}\tag{9}$$

where 1 is the 2×2 identity matrix, and Z_i appears at the i -th place (since it only addresses the i -th qubit).

In terms of these single-qubit unitaries, factoring equation (8) for each qubit we obtain N equations:

$$\begin{aligned}U_1 &= C'_N C_N \cdots C_2 Z_1 C_1, \\ U_2 &= C'_N C_N \cdots Z_2 C_2 C_1, \\ &\vdots \\ U_N &= C'_N C_N \cdots C_2 C_1.\end{aligned}\tag{10}$$

From the last equation we can determine $C'_N C_N$:

$$C'_N C_N = U_N C_1^{-1} C_2^{-1} \cdots C_{N-1}^{-1},\tag{11}$$

and eliminating this factor from the remaining equations we obtain:

$$\begin{aligned}U_N^{-1} U_1 &= C_1^{-1} Z_1 C_1, \\ U_N^{-1} U_2 &= C_1^{-1} C_2^{-1} Z_2 C_2 C_1, \\ &\vdots \\ U_N^{-1} U_{N-1} &= C_1^{-1} C_2^{-1} \cdots C_{N-1}^{-1} Z_{N-1} C_{N-1} \cdots C_2 C_1.\end{aligned}\tag{12}$$

We solve each equation in (12) consecutively. To solve the first equation in (12), let us notice that its left-hand side is a known unitary, which can be written as:

$$U_N^{-1} U_1 = e^{-i\alpha_1 u_1/2},\tag{13}$$

where α_1 is the angle of the rotation and u_1 its generator. The right-hand side is simply a rotation around Z and a change of basis. Therefore, the rotation angle of Z_1 must be equal to α_1 , and the change of basis must be such that:

$$u_1 = C_1^{-1} \sigma_z C_1.\tag{14}$$

We show in appendix A.1 how to find the generator and angle of the collective rotation C_1 .

Having determined C_1 , we can write the second equation in (12) as:

$$C_1 U_N^{-1} U_2 C_1^{-1} = C_2^{-1} Z_2 C_2.\tag{15}$$

As before, the left-hand side of this equation is a known unitary, and the right-hand side consists of a rotation around Z and a change of basis, so the rotation angle θ_2 and generator of the change of basis C_2 can be found as for the previous equation. This procedure can be repeated until all of the C_k and Z_k with $k \leq N - 1$ are determined. The last collective operations C_N and C'_N can be determined from equation (11). For this we need to decompose an arbitrary unitary into a product of two equatorial rotations; this can be done as explained in appendix A.2.

We have shown so far how to compile a local unitary exactly. However, in certain cases the constraints on the target unitary are weaker, so that it can be implemented with a simpler sequence. For instance, a unitary that is followed by global gates whose phase can be freely adjusted must only be specified up to a collective Z rotation afterwards, since this rotation can be absorbed into the phase. This removes one free parameter from the sequence, thus simplifying its implementation. The details of this procedure are presented in appendix A.3. Another case of interest is when the target unitary is specified up to arbitrary independent Z rotations afterwards, for instance when the unitary is followed by a projective measurement on the Z basis. This is particularly useful for tomographic measurements; details are shown in appendix A.4.

3. Compilation of general unitaries

In section 2 we studied how to compile local unitaries in terms of collective and addressed rotations. However, a universal quantum computer also requires entangling unitaries, which must be compiled into the experimentally available local and entangling gates. For example, in figure 2 we show a decomposition of a Toffoli gate into a sequence of local and entangling gates applied consecutively. In this section, we present an algorithm to find such decompositions for arbitrary unitaries.

We seek decompositions directly in terms of multi-qubit entangling gates, since these are often more efficient than decompositions in terms of two-qubit gates. For example, a Toffoli gate can be implemented using

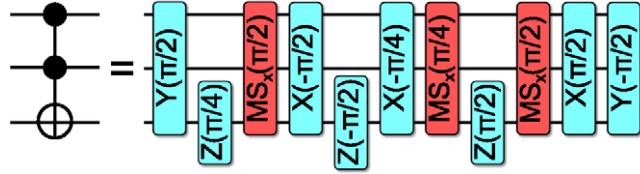


Figure 2. Decomposition of a Toffoli gate into a pulse sequence of collective equatorial rotations, addressed Z rotations and entangling Mølmer-Sørensen (MS) gates.

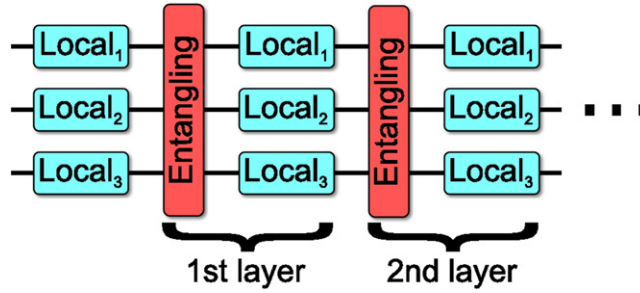


Figure 3. Sequence with layers of local and entangling gates applied consecutively.

only 3 Mølmer-Sørensen (MS) gates [11], while 6 CNOT gates are needed to implement it [21]), and a Fredkin gate can be implemented using 4 MS gates [22], while the least number of two-qubit gates required is 5 [23]. As described in section 1.1, many equivalent types of entangling gates are experimentally available. We will consider MS gates, but the methods shown here are applicable to any entangling gate that forms a universal set together with local operations.

3.1. Compilation in layers

In many quantum information processing experiments the most costly operations in terms of fidelity are entangling gates. Therefore, when trying to compile a unitary we seek to minimize the number of those. A straightforward way to do this is to use pulse sequences where layers of local unitaries and entangling gates are applied consecutively, as shown in figure 3.

Any unitary can be decomposed in terms of single-qubit gates and two-qubit CNOT gates [24]:

$$U = L_M \text{CNOT}_M \cdots L_1 \text{CNOT}_1 L_0, \quad (16)$$

where L_i denotes an arbitrary local unitary on the whole qubit register and CNOT_i denotes a gate between some two qubits. A two-qubit CNOT gate can be implemented in an arbitrary N -qubit register as a sequence of local unitaries and $\text{MS}_x(\pi/8)$ gates [11]. Therefore, the following decomposition is always possible:

$$U = L_M \text{MS}_x(\pi/8) \cdots L_1 \text{MS}_x(\pi/8) L_0. \quad (17)$$

However, some of the local unitaries L_i in a decomposition of the form (17) may actually be identity, so after removing them the resulting sequence has the following structure:

$$U = L_M \text{MS}_x(k_M\pi/8) \cdots L_1 \text{MS}_x(k_1\pi/8) L_0, \quad (18)$$

where the k_i are integers, and the number M of entangling gates is the same or less than in equation (17). It is enough to consider $0 \leq k \leq 7$, since $\text{MS}_x(\pi)$ is either the identity for an odd number of qubits, or a π rotation around X for an even number of qubits.

We now seek to further simplify sequence (18). Every single-qubit unitary U_i on qubit i can be written as a composition of rotations around two different fixed axes [6], which means that we can always choose α_{i1} , α_{i2} and α_{i3} such that:

$$U_i = X_i(\alpha_{i3})Z_i(\alpha_{i2})X_i(\alpha_{i1}). \quad (19)$$

Any local unitary $L = \prod_{i=1}^N U_i$ can therefore be written as:

$$L = \prod_{i=1}^N X_i(\alpha_{i3})Z_i(\alpha_{i2})X_i(\alpha_{i1}), \quad (20)$$

where the product goes over the N qubits in the register. Since unitaries acting on different qubits commute, we can write this as:

$$L = \prod_{i=1}^N X_i(\alpha_{i3}) \prod_{i=1}^N Z_i(\alpha_{i2}) \prod_{i=1}^N X_i(\alpha_{i1}) \quad (21)$$

$$= \tilde{X}' \tilde{Z} \tilde{X}, \quad (22)$$

where \tilde{X} and \tilde{Z} denote arbitrary products of rotations around the X or Z axes for all qubits. Therefore, the sequence in (18) can be written as:

$$U = \tilde{X}'_M \tilde{Z}_M \tilde{X}_M \text{MS}_x(k_M \pi/8) \cdots \\ \times \tilde{X}'_1 \tilde{Z}_1 \tilde{X}_1 \text{MS}_x(k_1 \pi/8) \tilde{X}'_0 \tilde{Z}_0 \tilde{X}_0, \quad (23)$$

and commuting the X rotations with the MS gates we obtain a sequence of the form:

$$U = \tilde{X}'_M \tilde{Z}_M \tilde{X}_M \text{MS}_x(k_M \pi/8) \cdots \\ \times \tilde{X}'_2 \tilde{Z}_2 \tilde{X}_2 \text{MS}_x(\alpha_2) \tilde{Z}_1 \text{MS}_x(k_1 \pi/8) \tilde{X}'_0 \tilde{Z}_0 \tilde{X}_0. \quad (24)$$

Every odd local unitary (except for the last one) is a product of Z rotations on all qubits, and the even local unitaries can be grouped as $L_i = \tilde{X}'_i \tilde{Z}_i \tilde{X}_i$. Moreover, a collective Z rotation can be extracted from each even local unitary L_i and absorbed into the phase of the subsequent MS gates and collective operations to simplify the implementation of L_i . Therefore the sequence can be written as:

$$U = L_M \text{MS}_{\phi_M}(k_M \pi/8) \cdots \\ \times L_2 \text{MS}_{\phi_2}(k_2 \pi/8) \tilde{Z}_1 \text{MS}_{\phi_1}(k_1 \pi/8) L_0. \quad (25)$$

We have thus shown that any N -qubit unitary U can be decomposed into a sequence of the form shown in (25). These sequences always have the same structure, which makes it easier to identify patterns if one wants to compile families of unitaries, i.e. unitaries that depend on some tunable parameter.

3.2. Numerical optimization

We have described a general form of a sequence of local operations and global entangling gates that implements any desired target unitary. It remains to find the actual sequence parameters, that is, the rotation angles and phases of the gates. However, we do not know *a priori* how many entangling gates will be needed for a given unitary. Therefore we suggest the following algorithm:

1. Propose a sequence with $M = 0$ entangling gates.
2. Search numerically for the sequence parameters that maximize the fidelity with the target unitary.
3. If the sequence has converged to the desired unitary (i.e. the fidelity equals 1), stop. Otherwise increase M by 1 and go back to step 2.

When performing the numerical optimization in step 2 there might be a number of local optima in addition to the true global optimum, making fully deterministic optimization methods difficult to apply. We apply therefore a *repeated local search*, where an efficient deterministic optimization method is iterated, each time using randomly determined initial conditions. The initial conditions are chosen randomly for every optimization run, as experience has shown us that starting close to previously found local minima does not offer any improvement. The search is finalized whenever the fidelity with the target unitary is above some predefined threshold, or when a maximum number of tries is exceeded. An advantage of this method is that, since each optimization run starts from random initial conditions, these are easy to perform in parallel.

The algorithm chosen for each numerical optimization is the quasi-Newton method of Broyden, Fletcher, Goldfarb, and Shanno (BFGS) [25]. The function to be maximized is the fidelity of the unitary resulting from the pulse sequence with the target unitary. Since we are interested in exact solutions, we reject those solutions whose fidelity (normalized to the maximum value possible) is not equal to 1 within some tolerance threshold (usually 1%). The gradient of the fidelity can be calculated analytically as a function of the sequence parameters, which speeds up the computation as compared to using several evaluations of the fidelity function.

A previously used approach to this optimization problem was a combination of local gradient descent and simulated annealing (SA) [11], which also helps to avoid local maxima. However, this method did not make use of the analytic expression for the fidelity gradient, which speeds up the search. Moreover, its performance depends on the ‘topography’ of the optimization space and requires manual tuning of the search parameters to achieve optimal results. We have compared the BFGS and simulated annealing approaches by compiling 100 random unitaries uniformly distributed in the Haar measure as explained in [26] for different numbers of qubits. In our experience,

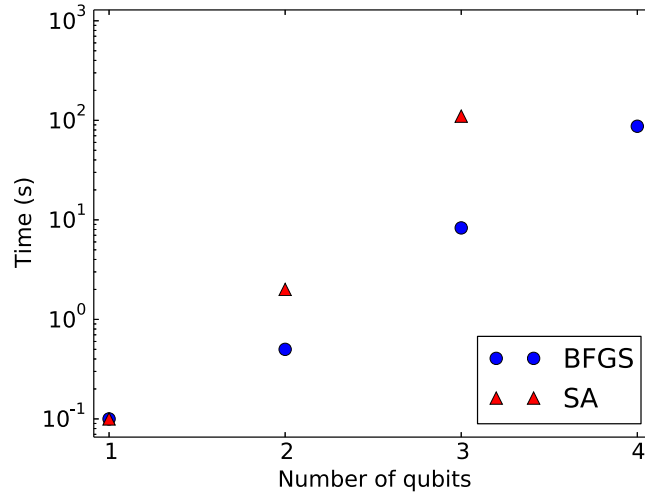


Figure 4. Average time required to find the global optimum for 100 unitaries randomly distributed in the Haar measure with the BFGS and simulated annealing methods, using an Intel®Core i5-4670s CPU 550 @ 3.10 GHz x 4 (one processing thread per optimization run). No data was obtained for the simulated annealing approach for 4 qubits owing to the excessive time required.

we find that the BFGS method scales better with the number of qubits than simulated annealing (see figure 4). The median number of search repetitions needed to find the global optimum was 1 in all the cases.

The exponential scaling of the optimization problem complexity depends on the number of entangling gates required to compile a given unitary, which is an intrinsic property of the unitary and does not depend on the search algorithm. It is already known that it is not possible to efficiently implement an arbitrary unitary in terms of two-qubit gates [6]; our numerical results suggest a similar result for N -qubit gates. In the two-qubit case the compilation always succeeded with 3 entangling gates, and not less (using 200 search repetitions). This was to be expected, since for two qubits an MS gate is equivalent to a CNOT gate, and it is known that 3 CNOT gates are enough (and in general necessary) to implement an arbitrary two-qubit unitary [27, 28]. In the three-qubit case, the optimization always succeeded with 8 entangling gates, and never with fewer (also using 200 repetitions). For 4 qubits, the optimization always succeeded for 25 entangling gates, and succeeded only 4% of the time with 24 entangling gates. However, we did only 4 optimization repeats in the four-qubit case, owing to the increased time it takes for these to converge. Therefore, it might be the case that given enough optimization runs, more unitaries would have been compiled with only 24 gates. We are not aware of any result in the literature concerning the number of N -qubit global entangling gates required for implementing a general N -qubit unitary for more than $N = 2$ qubits. From our numerical results, we conjecture that any three-qubit unitary can be implemented using at most 8 MS gates, and any four-qubit unitary using at most 24 or 25 MS gates.

A particularly interesting group of unitaries are Clifford gates, which find applications in quantum error correction [29], randomized benchmarking [30], and state distillation protocols [6]. To explore the difficulty of compiling such gates, we have tested our algorithm with randomly generated Clifford gates, as explained in [31]. We show in figure 5 the distribution of the optimal number of entangling gates required for compiling two-, three- and four-qubit unitaries. Our results agree with the literature [32] for the two-qubit case, since MS gates are then equivalent to controlled-Z (or CNOT) gates. For larger numbers of qubits, the performance of our algorithm in terms of number of multi-qubit gates required is also similar to that of algorithms based on two-qubit gates [32].

3.3. Compilation of isometries

A particular case of interest is the compilation of a unitary whose action only matters on certain input states. This happens, for instance, when one is interested in state preparation starting from some fixed input state. Such operations belong to the more general class of operations known as *isometries* [14, 30]. In this case, the problem to be solved has less constraints than when fully specifying the target unitary, so a simpler sequence may be found. In this section we focus on compiling a unitary that is only specified in a particular subspace of the input states, for example:

$$U_{\text{target}} = \begin{pmatrix} u_{11} & u_{12} & \vdots & \vdots \\ u_{21} & u_{22} & \text{free} & \text{free} \\ u_{31} & u_{32} & \vdots & \vdots \\ u_{41} & u_{42} & \vdots & \vdots \end{pmatrix}, \quad (26)$$

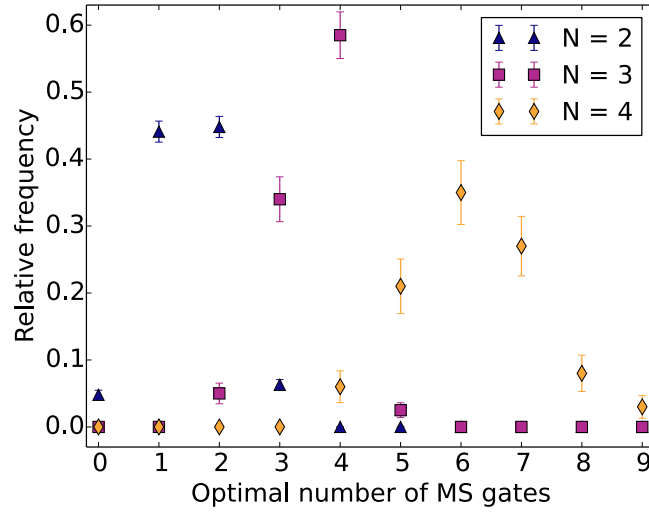


Figure 5. Optimal number of entangling gates needed to compile random Clifford operations, for sample sizes $\{1000, 200, 100\}$ for $N = \{2, 3, 4\}$ qubits. Each Clifford gate was generated using $10N^8$ steps of the random walk described in [31]. Error bars correspond to one standard deviation.

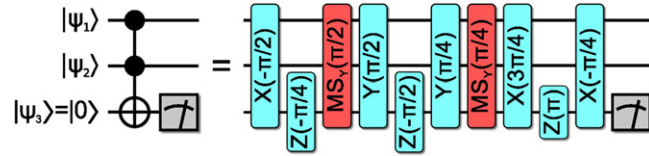


Figure 6. Left: a unitary mapping (Toffoli gate) is applied, after which qubit 3 is measured. Right: a pulse sequence for implementing the circuit on the left. This implementation is simpler than in the fully constrained case (figure 2) because of the additional degrees of freedom when compiling.

where the columns marked as ‘free’ are left unspecified. In this case, a suitable fidelity function for the numerical optimization is:

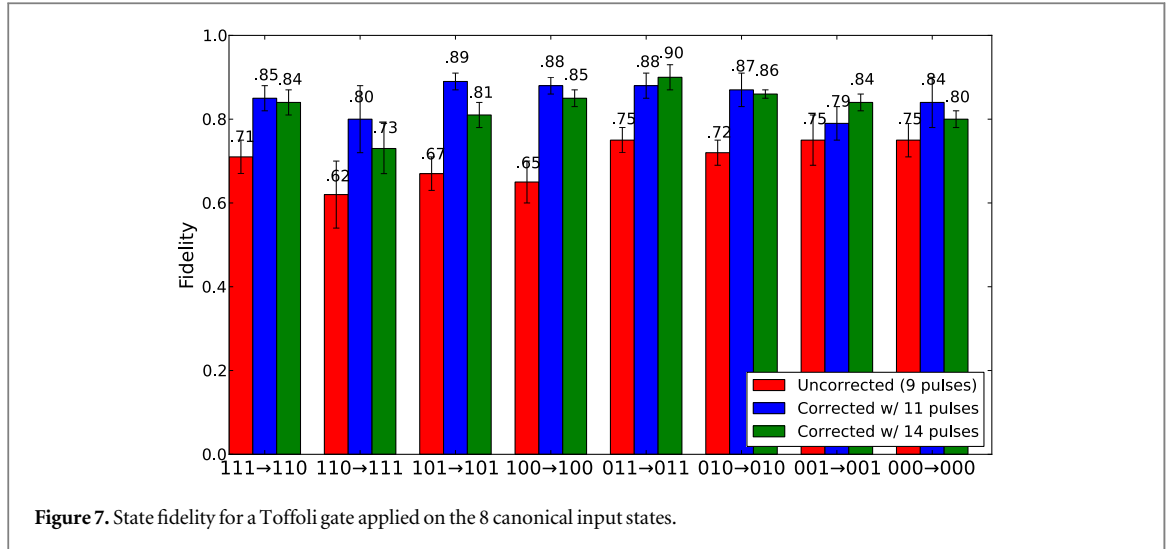
$$f(U) = |\text{tr}(U|_S U_{\text{target}}^\dagger|_S)|^2, \quad (27)$$

where $U|_S$ is a rectangular matrix with the components of the unitary in the restricted subspace.

A more general case is where some of the relative phases of the projections of the unitary acting on different subspaces of the whole Hilbert space are irrelevant. For example, suppose that one wants to apply a unitary to map some observable onto an ancilla qubit and then measure the ancilla, as shown in figure 6. Since the input state of qubit 3 is known to be $|0\rangle$, only the subspace of input states spanned by $\{|000\rangle, |010\rangle, |100\rangle, |110\rangle\}$ is relevant. Moreover, the measurement will project the state of the system onto either the subspace spanned by $\{|000\rangle, |010\rangle, |100\rangle\}$, or that spanned by $\{|111\rangle\}$, and all phase coherence between these alternatives will be lost. Therefore, the compiled sequence can be sought such that it matches the desired unitary in each of the subspaces but allowing an arbitrary phase ϕ between them:

$$U_{\text{target}} = \begin{pmatrix} 1 & \vdots & 0 & \vdots & 0 & \vdots & 0 & \vdots \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 & \vdots \\ 0 & \vdots & 1 & \vdots & 0 & \vdots & 0 & \vdots \\ 0 & \text{free} & 0 & \text{free} & 0 & \text{free} & 0 & \text{free} \\ 0 & \vdots & 0 & \vdots & 1 & \vdots & 0 & \vdots \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 & \vdots \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 & \vdots \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & e^{i\phi} & \vdots \end{pmatrix} \quad (28)$$

In this case (figure 6) it is possible to find a simpler implementation than in the fully constrained case (figure 2), owing to the additional degrees of freedom available, namely arbitrary outputs for the $|\psi_3\rangle = |1\rangle$ input states and an arbitrary relative phase between the two possible measurement outcomes.



In the general case considered here we want to maximize the fidelity in each subspace, without regard to the relative phases between these. Therefore we can seek to maximize the function f consisting of the sum of the fidelity functions (27) corresponding to each subspace:

$$f(U) = \sum_j |\text{tr}(U|_{S_j} U_{\text{target}}^\dagger|_{S_j})|^2, \quad (29)$$

where the sum goes over all the subspaces with different relative phases, and $U|_{S_j}$ is a rectangular matrix with the components of the unitary in the j -th subspace.

3.4. Compensation of systematic errors

Owing to systematic errors, the operations experimentally applied may still be unitary but deviate from the intended ones. An example of this is addressing crosstalk due to laser light leaking onto adjacent qubits. If it is possible to characterize the actual experimental operations being applied, then they can be taken into account for the compilation by adapting our optimization procedure:

1. Compile the target unitary in terms of the ideal gates.
2. Replace the ideal gates by the experimentally characterized operations.
3. Add operations to obtain a higher fidelity with the ideal target unitary.

As an example we show that excessive crosstalk can be corrected in an implementation of a Toffoli gate. Figure 7 depicts experimental data corresponding to the action of the Toffoli gate on the 8 input basis states. It can be seen that, by adding just two pulses, the output fidelity for each input state increased in some cases by up to 20%. The sequence with 11 pulses is actually only an approximate correction to the uncorrected case. The exact correction requires 14 pulses, and actually yields a lower fidelity than the approximate one, since it requires more pulses and each of these has a non-zero error probability.

4. Conclusions and outlook

In this work we have shown methods to compile quantum unitaries into a sequence of collective rotations, addressed rotations and global entangling operations. For local unitaries, we have demonstrated an analytic approach that produces the shortest possible sequences in the general case, and adapted the method to simplify the resulting sequences if some constraints on the unitary are lifted. For arbitrary unitaries, we have presented an approach that produces sequences of layered local and entangling operations. This approach is based on a numerical optimization procedure that is faster than previously used ones, and the sequences obtained are by design optimal with respect to the number of entangling gates. Our numerical results suggest upper bounds on the number of N -qubit gates required to implement arbitrary three- and four-qubit unitaries.

The results of this paper show that in many cases one may obtain more efficient implementations by considering operations more general than two-qubit entangling gates. However, the exponentially growing complexity of decompositions as the number of qubits increases points to the necessity of keeping the register size small.

Acknowledgments

We thank I Chuang, B Lanyon, and V Nebendahl for fruitful discussions. We thank the referees for bringing [14, 30–32] to our attention. We gratefully acknowledge support by the Austrian Science Fund (FWF), through the SFB FoQuS (FWF Project No. F4002-N16), as well as the Institut für Quantenoptik und Quanteninformation GmbH. EAM is a recipient of a DOC fellowship from the Austrian Academy of Sciences. PS was supported by the Austrian Science Foundation (FWF) Erwin Schrödinger Stipendium 3600-N27. This research was funded by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), through the Army Research Office grant W911NF-10-1-0284. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of IARPA, the ODNI, or the US Government.

Appendix A. Compiling local unitaries

A.1. Finding basis changes

In this appendix we will show how to satisfy equation (14). We need to find a rotation C around the equator of the Bloch sphere such that:

$$u = C^{-1}\sigma_z C, \quad (\text{A1})$$

where u is the generator of a given known unitary U , and it can always be written as:

$$u = \sin \theta \cos \phi \sigma_x + \sin \theta \sin \phi \sigma_y + \cos \theta \sigma_z, \quad (\text{A2})$$

for some angles θ, ϕ .

In general C is of the form:

$$C = e^{-i\gamma c/2}, \quad (\text{A3})$$

where γ is its rotation angle and c its generator, which must lie on the equator and thus be a linear combination of σ_x and σ_y . If we propose:

$$c = \sin \phi \sigma_x - \cos \phi \sigma_y, \quad (\text{A4})$$

and replace in equation (A1), we find that the angle of rotation must be:

$$\gamma = \theta. \quad (\text{A5})$$

A.2. Writing a unitary as a product of two equatorial rotations

We will show here how to decompose an arbitrary unitary as a product of two rotations around the equator of the Bloch sphere, namely:

$$U = C_2 C_1. \quad (\text{A6})$$

The target unitary can be written as:

$$U = \cos\left(\frac{\beta}{2}\right)1 - i \sin\left(\frac{\beta}{2}\right) \times (\sin \theta \cos \phi \sigma_x + \sin \theta \sin \phi \sigma_y + \cos \theta \sigma_z), \quad (\text{A7})$$

where β is its rotation angle, and θ, ϕ determine its rotation axis. Similarly, the equatorial rotations can be written as:

$$C_i = \cos\left(\frac{\alpha_i}{2}\right)1 - i \sin\left(\frac{\alpha_i}{2}\right)(\cos \phi'_i \sigma_x + \sin \phi'_i \sigma_y), \quad (\text{A8})$$

for some rotation angles α_i and phases ϕ'_i .

We shall assume that:

$$\alpha_1 = \alpha_2 = \alpha, \quad (\text{A9})$$

$$\phi'_1 = \phi + \Delta/2, \quad (\text{A10})$$

$$\phi'_2 = \phi - \Delta/2. \quad (\text{A11})$$

Replacing these into (A6) and solving for α and Δ we obtain:

$$\cos^2\left(\frac{\alpha}{2}\right) = \frac{1}{2}\left(\cos\left(\frac{\beta}{2}\right) + 1\right)\sin^2 \theta, \quad (\text{A12})$$

$$\cos \Delta = \frac{\cos^2\left(\frac{\alpha}{2}\right) - \cos\left(\frac{\beta}{2}\right)}{1 - \cos^2\left(\frac{\alpha}{2}\right)}. \quad (\text{A13})$$

A.3. Unitaries up to a collective Z rotation

Suppose that the unitary U we want to implement is followed by gates whose phase can be freely chosen. Then it must only be specified up to an arbitrary collective rotation Z' , since this phase can be absorbed in the following gates. To compile U , we shall consider a decomposition of the form (7):

$$U = Z' C_N Z_{N-1} C_{N-1} \cdots Z_2 C_2 Z_1 C_1. \quad (\text{A14})$$

Such a decomposition is more convenient in this case because the last addressed pulse Z_N has been eliminated by taking advantage of the additional degree of freedom provided by Z' . We can now follow the same steps as in section 2. The unitary C_N is given by:

$$C_N = Z'^{-1} U_N C_1^{-1} C_2^{-1} \cdots C_{N-1}^{-1}, \quad (\text{A15})$$

and eliminating this factor from the rest of the equations we obtain:

$$\begin{aligned} U_N^{-1} U_1 &= C_1^{-1} Z_1 C_1, \\ U_N^{-1} U_2 &= C_1^{-1} C_2^{-1} Z_2 C_2 C_1, \\ &\vdots \\ U_N^{-1} U_{N-1} &= C_1^{-1} C_2^{-1} \cdots C_{N-1}^{-1} Z_{N-1} C_{N-1} \cdots C_2 C_1. \end{aligned} \quad (\text{A16})$$

Equation (A16) can be satisfied in exactly the same way as explained in section 2. In order to satisfy equation (A15) we need to find a rotation Z' such that the generator of C_N lies on the equator. This can be done as follows.

We wish to find how to satisfy equation (A15). For this we need to find a rotation Z around the Z axis and a rotation C around an axis on the equator of the Bloch sphere such that, for a given unitary U , the following equation holds:

$$C = ZU. \quad (\text{A17})$$

U is in general of the form:

$$U = e^{-i\alpha u/2}, \quad (\text{A18})$$

and Z is of the form:

$$Z = e^{-i\beta \sigma_z/2}. \quad (\text{A19})$$

We will first find the angle of rotation β . If we write out (A17) in terms of the generators of U and Z we have:

$$\begin{aligned} C &= \left(\cos\left(\frac{\beta}{2}\right) 1 - i \sin\left(\frac{\beta}{2}\right) \sigma_z \right) \\ &\quad \times \left(\cos\left(\frac{\alpha}{2}\right) 1 - i \sin\left(\frac{\alpha}{2}\right) u \right). \end{aligned} \quad (\text{A20})$$

Since the axis of rotation of C lies on the equator, its generator must not have any Z component, and thus:

$$0 = \sin\left(\frac{\beta}{2}\right) \cos\left(\frac{\alpha}{2}\right) + \cos\left(\frac{\beta}{2}\right) \sin\left(\frac{\alpha}{2}\right) u_z, \quad (\text{A21})$$

that is:

$$\beta = -2 \arctan\left(\tan\left(\frac{\alpha}{2}\right) u_z\right). \quad (\text{A22})$$

Once β is known, the unitary on the right-hand side of (A17) is fully determined, and thus C as well.

A.4. Unitaries up to independent Z rotations

Finally, suppose that the unitary we want to implement is defined up to arbitrary independent rotations for each qubit around the Z axis. This is useful if the unitary is followed by a projective measurement, since any final rotation around the measurement axis for any qubit simply adds a phase and will not change the measured probabilities.

Let us again consider a sequence of the form (7). The decomposition must now satisfy, for each qubit:

$$\begin{aligned} Z'_1 U_1 &= C_N \cdots C_2 Z_1 C_1, \\ Z'_2 U_2 &= C_N \cdots Z_2 C_2 C_1, \\ &\vdots \\ Z'_N U_N &= Z_N C_N \cdots C_2 C_1, \end{aligned} \quad (\text{A23})$$

where the Z'_i are arbitrary rotations around the Z axis. As before, we can set $Z_N = 1$ and find C_N :

$$C_N = Z'_N U_N C_1^{-1} C_2^{-1} \cdots C_{N-1}^{-1}. \quad (\text{A24})$$

Eliminating C_N from the remaining equations we obtain:

$$\begin{aligned} U_N^{-1} Z_N'^{-1} Z'_1 U_1 &= C_1^{-1} Z_1 C_1, \\ U_N^{-1} Z_N'^{-1} Z'_2 U_2 &= C_1^{-1} C_2^{-1} Z_2 C_2 C_1, \\ &\vdots \\ U_N^{-1} Z_N'^{-1} Z'_{N-1} U_{N-1} &= C_1^{-1} \cdots C_{N-1}^{-1} Z_{N-1} C_{N-1} \cdots C_1. \end{aligned} \quad (\text{A25})$$

Each equation has now an extra degree of freedom coming from the angle of the Z'_k rotation. Let us for simplicity consider the case where the number of qubits N is odd. If we group equation (A25) in pairs we get two degrees of freedom per pair, which can be used to remove one of the global operations. Therefore we will discard every even-numbered global operation C_{2k} from our decomposition and look for the solution of the following system of equations:

$$\begin{aligned} U_N^{-1} Z_1'' U_1 &= C_1^{-1} Z_1 C_1, \\ U_N^{-1} Z_2'' U_2 &= C_1^{-1} Z_2 C_1, \\ U_N^{-1} Z_3'' U_3 &= C_1^{-1} C_3^{-1} Z_3 C_3 C_1, \\ U_N^{-1} Z_4'' U_4 &= C_1^{-1} C_3^{-1} Z_4 C_3 C_1, \\ &\vdots \\ U_N^{-1} Z_{N-2}'' U_{N-2} &= C_1^{-1} \cdots C_{N-2}^{-1} Z_{N-2} C_{N-2} \cdots C_1, \\ U_N^{-1} Z_{N-1}'' U_{N-1} &= C_1^{-1} \cdots C_{N-2}^{-1} Z_{N-1} C_{N-2} \cdots C_1, \end{aligned} \quad (\text{A26})$$

where $Z_k'' = Z_N'^{-1} Z'_k$. If the number of qubits N is even, then the last equation is simply left unpaired. It is easy to verify that for each pair of equations the right-hand sides commute, and therefore we must have:

$$[U_N^{-1} Z_{2k-1}'' U_{2k-1}, U_N^{-1} Z_{2k}'' U_{2k}] = 0, \quad (\text{A27})$$

or equivalently:

$$[Z_{2k-1}'' U_{2k-1} U_N^{-1}, Z_{2k}'' U_{2k} U_N^{-1}] = 0. \quad (\text{A28})$$

In order to solve equation (A28) we need to find rotations $Z_1 = Z(\beta_1)$, $Z_2 = Z(\beta_2)$ that satisfy a general equation of the form:

$$[Z_1 U_1, Z_2 U_2] = 0, \quad (\text{A29})$$

for given arbitrary U_1 , U_2 , whose generators are u_1 and u_2 respectively.

Let us define:

$$V_i = Z_i U_i, \quad (\text{A30})$$

and let v_i be the generators of the V_i . In order to satisfy (A29), the v_i must satisfy:

$$v_1 = v_2 = v, \quad (\text{A31})$$

since if two unitaries commute their generators must be the same. Our first goal is to determine the generator v .

Let us consider the unitary:

$$W_i = Z_i^{1/2} U_i Z_i^{1/2}. \quad (\text{A32})$$

By writing down W_i explicitly in terms of the generators of each factor, it can be seen that its generator w_i satisfies:

$$\{w_i, [\sigma_z, u_i]\} = 0. \quad (\text{A33})$$

Since we have:

$$V_i = Z_i^{1/2} W_i Z_i^{-1/2}, \quad (\text{A34})$$

from equation (A33) we see that:

$$\{v, Z_i^{1/2}[\sigma_z, u_i]Z_i^{-1/2}\} = 0. \quad (\text{A35})$$

The geometrical meaning of this equation is that the vector defined by v on the Bloch sphere is perpendicular to that defined by $Z_i^{1/2}[\sigma_z, u_i]Z_i^{-1/2}$. Since (A35) must hold for $i = 1, 2$, v must correspond to the cross product of these vectors:

$$v = \mathcal{N}[Z_1^{1/2}[z, u_1]Z_1^{-1/2}, Z_2^{1/2}[\sigma_z, u_2]Z_2^{-1/2}], \quad (\text{A36})$$

where \mathcal{N} is chosen such that:

$$\frac{1}{2}\text{tr}(v^2) = 1. \quad (\text{A37})$$

Having found v , it remains to find the rotation angles β_i . Now, v must satisfy $[Z_i U_i, v] = 0$, and therefore:

$$U_i v U_i^{-1} = Z(\beta_i)^{-1} v Z(\beta_i). \quad (\text{A38})$$

Both v and U_i are known, so v and $U_i v U_i^{-1}$ can be written down explicitly as:

$$v = \sin \theta \cos \phi \sigma_x + \sin \theta \sin \phi \sigma_y + \cos \theta \sigma_z, \quad (\text{A39})$$

$$U_i v U_i^{-1} = \sin \theta \cos \phi'_i \sigma_x + \sin \theta \sin \phi'_i \sigma_y + \cos \theta \sigma_z, \quad (\text{A40})$$

and therefore:

$$\beta_i = \phi - \phi'_i. \quad (\text{A41})$$

We have shown how to find suitable rotations Z'' that fulfill condition (A28). Once these are found, all the left-hand sides of (A26) are known unitaries and the system can be solved as before. The last collective rotation C_N can be determined from (A24) as shown in appendix A.3. We have thus shown how to compile the sought unitary U into a sequence of the form:

$$U = \begin{cases} C_N Z_{N-1} Z_{N-2} C_{N-2} \cdots C_3 Z_2 Z_1 C_1 & \text{for odd } N, \\ C_N Z_{N-1} C_{N-1} \cdots C_3 Z_2 Z_1 C_1 & \text{for even } N. \end{cases} \quad (\text{A42})$$

References

- [1] Ekert A and Jozsa R 1996 *Rev. Mod. Phys.* **68** 733
- [2] Gisin N, Ribordy G, Tittel W and Zbinden H 2002 *Rev. Mod. Phys.* **74** 145
- [3] Cirac J I and Zoller P 2012 *Nat. Phys.* **8** 264
- [4] Bloch I, Dalibard J and Nascimbène S 2012 *Nat. Phys.* **8** 267
- [5] Blatt R and Roos C F 2012 *Nat. Phys.* **8** 277
- [6] Nielsen M A and Chuang I L 2004 *Quantum Computation and Quantum Information* 1st edn (Cambridge: Cambridge University Press) pp 191–3
- [7] Schindler P et al 2013 *New J. Phys.* **15** 123012
- [8] Harty T P, Allcock D T C, Ballance C J, Guidoni L, Janacek H A, Linke N M, Stacey D N and Lucas D M 2014 *Phys. Rev. Lett.* **113** 220501
- [9] Xia T, Lichtman M, Maller K, Carr A W, Piotrowicz M J, Isenhower L and Saffman M 2015 *Phys. Rev. Lett.* **114** 100503
- [10] Vandersypen L M K and Chuang I L 2005 *Rev. Mod. Phys.* **76** 1037
- [11] Nebendahl V, Häffner H and Roos C F 2009 *Phys. Rev. A* **79** 012312
- [12] Khaneja N and Glaser S J 2001 *Chem. Phys.* **267** 11
- [13] Knill E 1995 *LANL Report* LAUR-95-22
- [14] Iten R, Colbeck R, Kululjan I, Home J and Christandl M 2016 *Phys. Rev. A* **93** 032318
- [15] Sørensen A and Mølmer K 2000 *Phys. Rev. A* **62** 22311
- [16] Debnath S, Linke N M, Figgatt C, Landsman K A, Wright K and Monroe C 2016 arXiv:1603.04512
- [17] Gaebler J P et al 2012 *Phys. Rev. Lett.* **108** 260503
- [18] Monroe C, Raussendorf R, Ruthven A, Brown K R, Maunz P, Duan L-M and Kim J 2014 *Phys. Rev. A* **89** 022317
- [19] Ballance C J, Harty T P, Linke N M, Sepiol M A and Lucas D M 2015 1 arXiv:1512.04600
- [20] Isenhower L, Saffman M and Molmer K 2011 *Quantum Inf. Process* **10** 755
- [21] Shende V V and Markov I L 2009 *Quant. Inf. Comp.* **9** 461
- [22] Monz T, Nigg D, Martinez E A, Brandl M F, Schindler P, Rines R, Wang S X, Chuang I L and Blatt R 2016 *Science* **351** 1068
- [23] Yu N and Ying M 2013 arXiv:1301.3727
- [24] DiVincenzo D P 1995 *Phys. Rev. A* **51** 1015
- [25] Nocedal J and Wright S J 1999 *Numerical Optimization* (New York: Springer)
- [26] Mezzadri F 2007 *Not. Am. Math. Soc.* **54** 592
- [27] Vatan F and Williams C 2004 *Phys. Rev. A* **69** 032315
- [28] Hanneke D, Home J P, Jost J D, Amini J M, Leibfried D and Wineland D J 2009 *Nat. Phys.* **6** 13
- [29] Gottesman D 1999 *Proc. XXII Int. Colloq. Gr. Theor. Methods Phys.* ed S P Corney, R Delbourgo and P D Jarvis (Cambridge, MA: International Press) 32–43
- [30] Knill E, Leibfried D, Reichle R, Britton J, Blakestad R B, Jost J D, Langer C, Ozeri R, Seidelin S and Wineland D J 2008 *Phys. Rev. A* **77** 012307
- [31] DiVincenzo D P, Leung D W and Terhal B M 2002 *IEEE Trans. Inf. Theory* **48** 580
- [32] Kliuchnikov V and Maslov D 2013 *Phys. Rev. A* **88** 052307